ROSE Compiler Infrastructure

Source-to-Source Analysis and Optimization

Dan Quinlan

Rich Vuduc, Qing Yi, Markus Schordan

Center for Applied Scientific Computing

Lawrence Livermore National Laboratory

This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract W-7405-Eng-48.

Overview

- ROSE Compiler Infrastructure
- Research Objectives
  - General Optimization of existing applications
  - Optimization of High-Level Abstractions
    - Telescoping Language
    - Plus/Minus Languages
    - Many other names for this
  - Empirical Optimization
- Targets non-compiler audience
- Emphasis on whole program capabilities
- Open Source (EDG part as binary)
- Conclusions

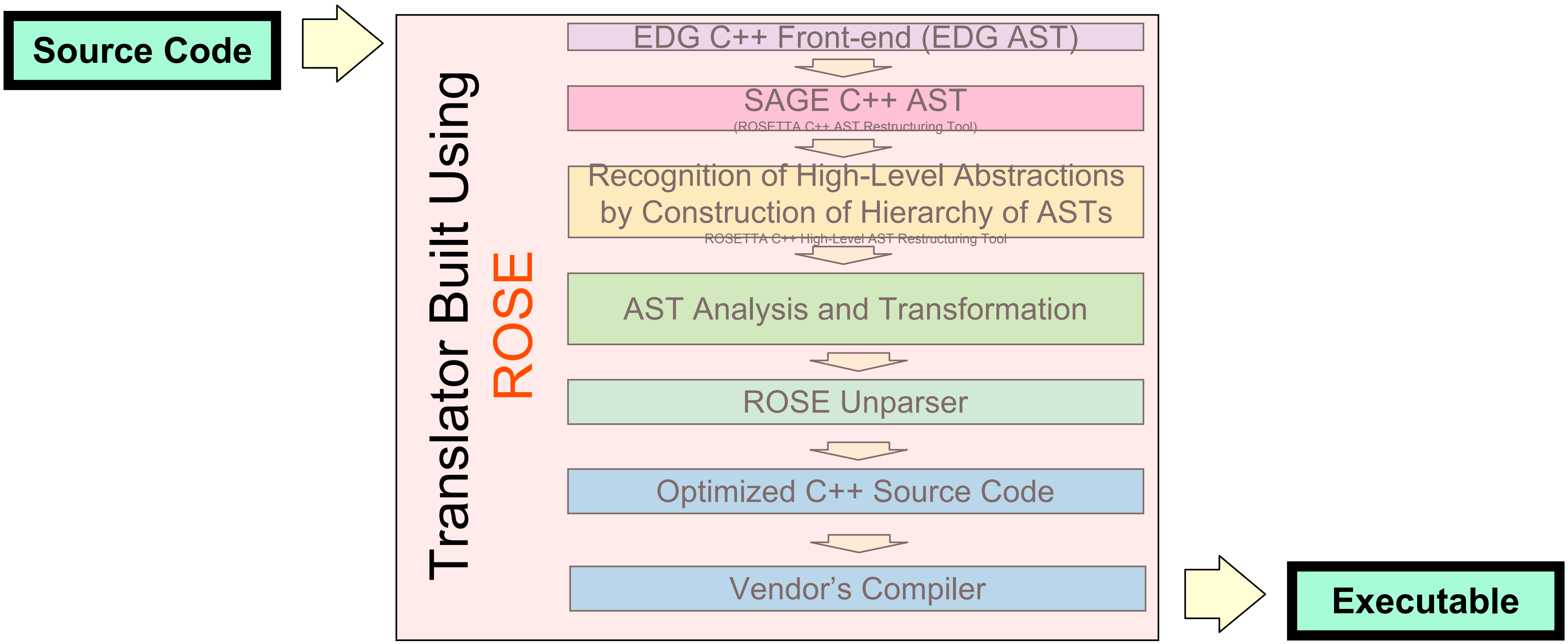
Motivation for Compiler Based Tools

- Current Status:
  - DOE generates huge amounts of software
  - ROSE provides a mechanism to automatically read, analyze, and fully rewrite ASC scale software in C, C++ (and eventually F90 as part of collaboration with Rice, we hope).
- ROSE Project focus IS on *optimization*

But a lot of tools could be built …,

Simple tools can only discover superficial things about software, to really know what is going on in an application you need a compiler infrastructure.

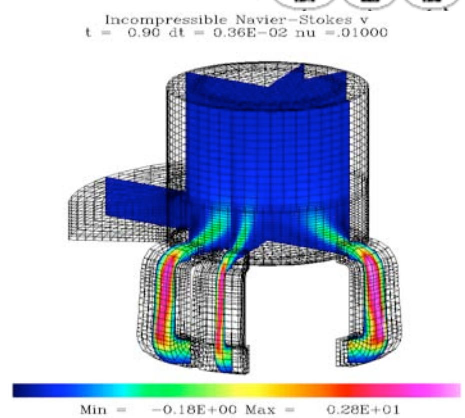
ROSE Source-to-Source Approach



- **ROSE Translator acts just like the vendor compiler**
- **Replaces compiler in application's *Makefile***

ROSE Project

- Software analysis and optimization for scientific applications
- Tool for building source-to-source translators
- Support for C and C++
- F90 in development
- Loop optimizations
- Lab and academic use
- Software engineering
- Performance analysis
- Domain-specific analys
- Development of new op
- Optimization of object-



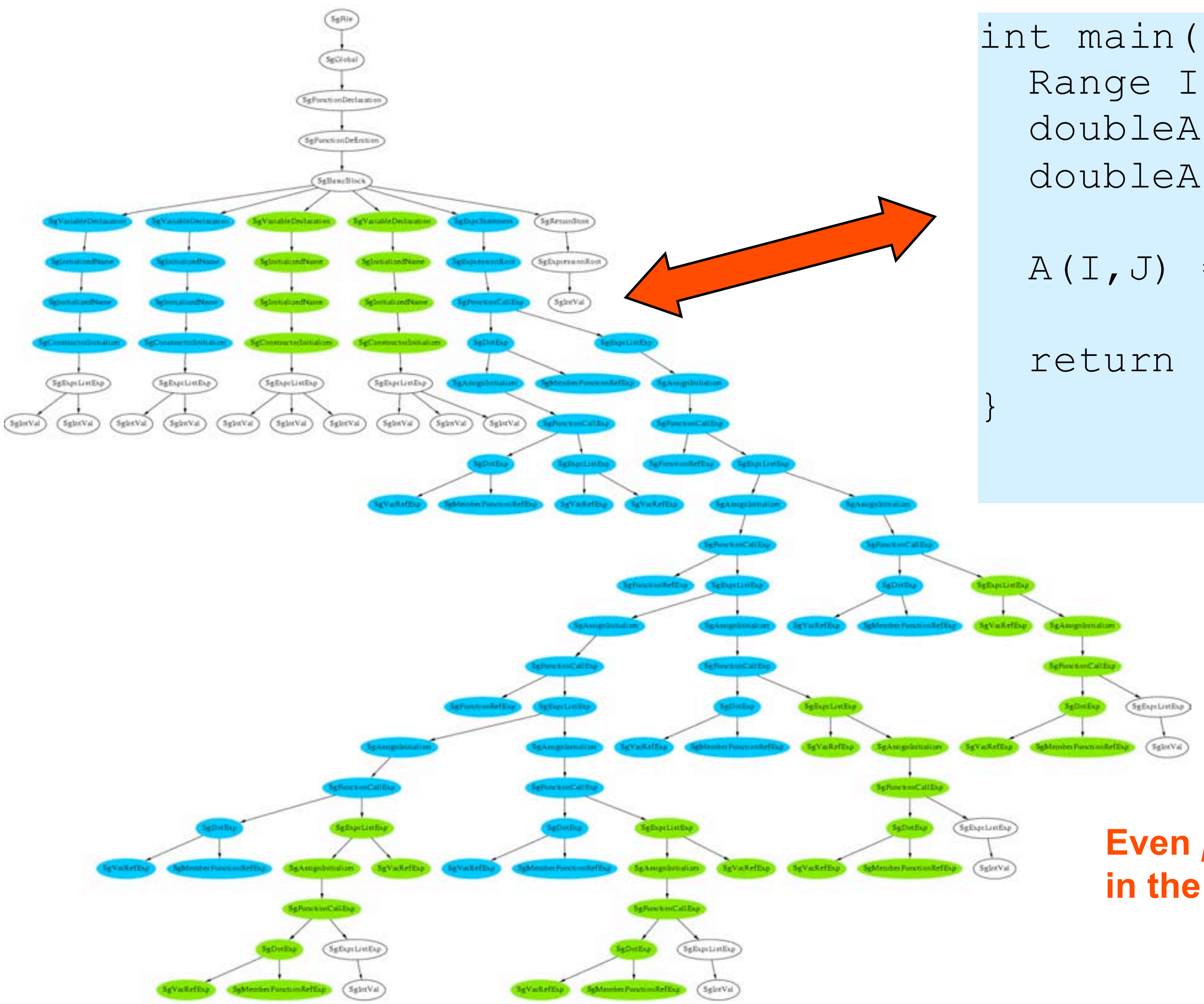
CASC

Program Analysis and Optimization

- Program Analysis (most are from Qing + contributions from students)
  - Call graph
    - Resolution of function pointers
    - Resolution of Virtual functions
    - Resolution of pointers to virtual functions
  - Dependence (procedural)
  - Control Flow (working on inter-procedural case)
  - Slicing (inter-procedural)
  - Partial Redundancy Elimination (PRE)
  - Connection to Open Analysis (work with ANL)
- Optimizations
  - Loop Optimization (Qing Yi)
    - Loop fusion, fission, blocking, unroling, array copy, etc.
    - inlining and outlining
  - Annotation Based Optimizations
  - Custom optimizations
    - Define your own optimization (high level or low level)

Automated Recognition

(Library Abstractions and other things)



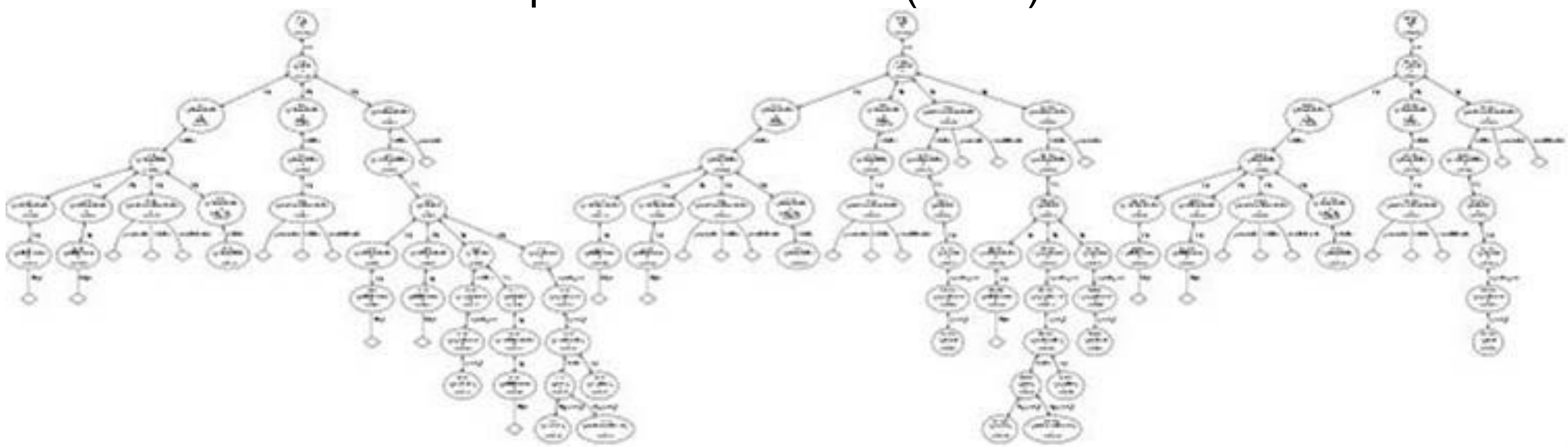
```
int main() {
  Range I(1,98,1),J(1,98,1);
  doubleArray A(100,100);
  doubleArray B(100,100);

  A(I,J) = B(I+1,J)+B(I-1,J)
          +B(I,J+1)+B(I,J-1);
  return 0;
}
```

Even *performance* information is in the AST (HPC Toolkit, RICE)

ROSE Whole Application Analysis

Three Separate source files (ASTs)



Supports Whole Program Analysis (Alternative to SQLite Interface)

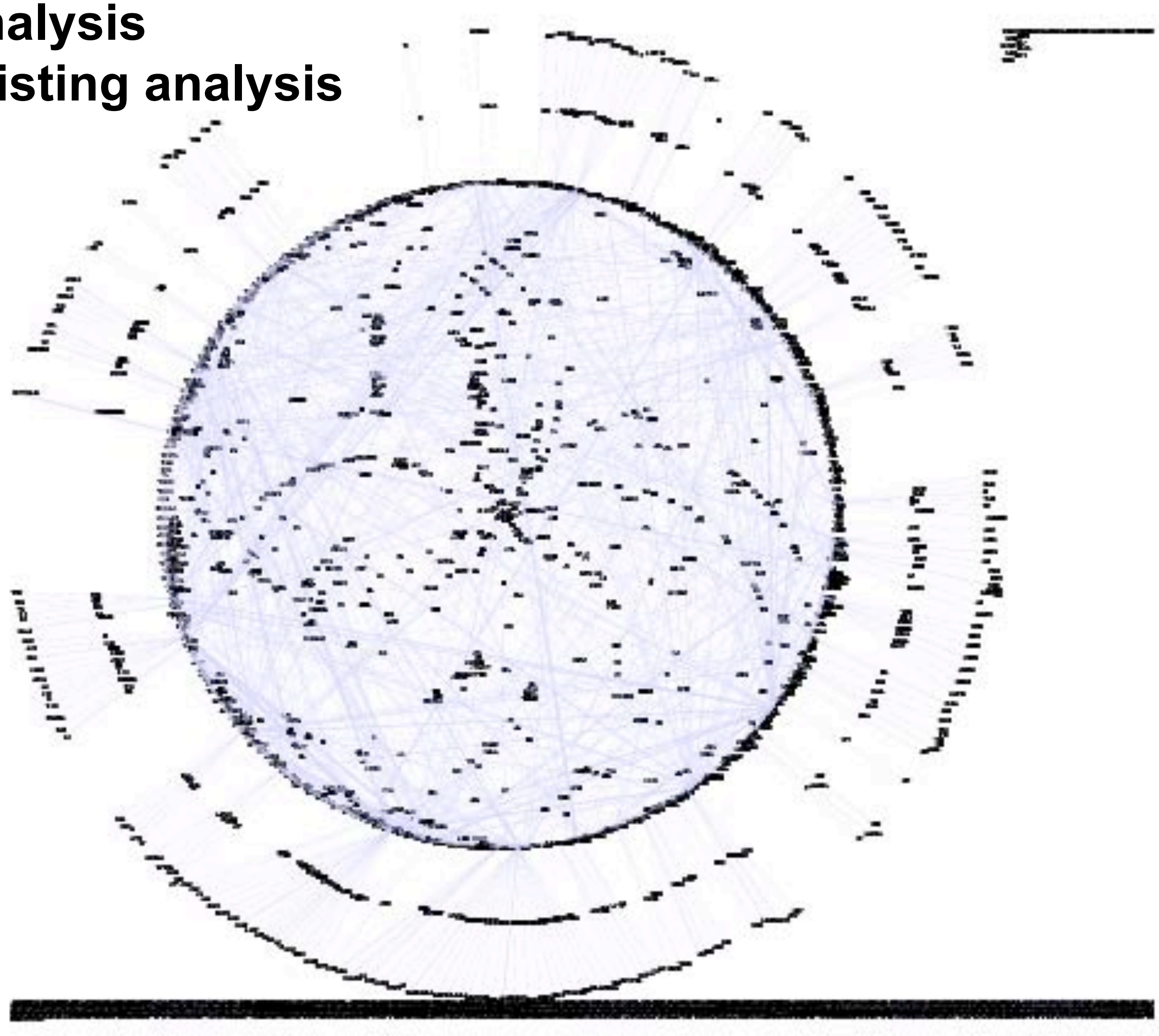
- **Shares AST nodes**
- **Preserves simplicity**
- **Preserves all analysis info**
- **Simple tools work on whole ASC applications**
- **Supports hundreds of source files**
- **Supports million line applications**

CASC

Merged ASTs save space and permit whole ASC scale application analysis

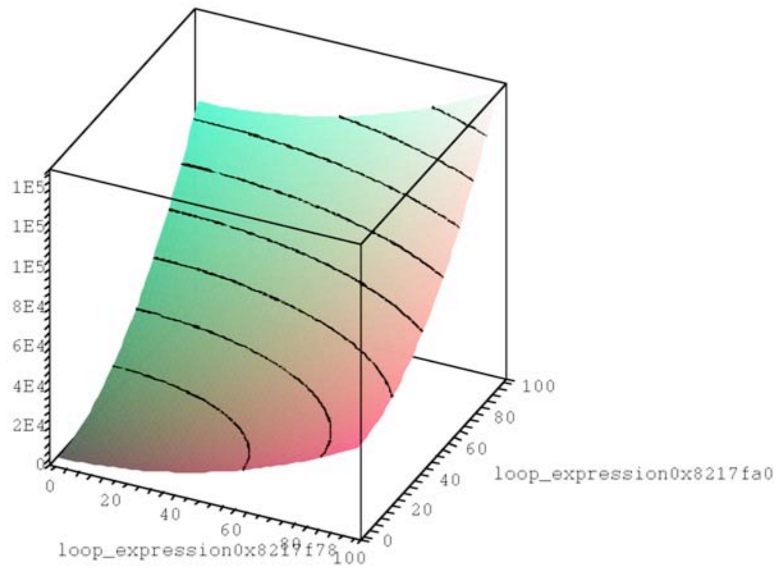
Large-Scale Application Support

- Call Graph Analysis
- Scaling up existing analysis

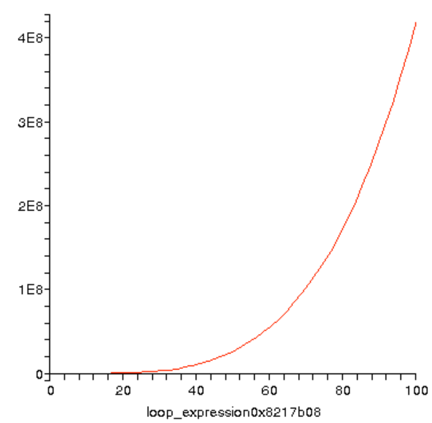


Automated Generation of Symbolic Equations for building Application Models

```
// Count -controlled loop complexity:
// 19+13*loop_expression0x8217f78+4*loop_expression0x8217f78^2+
//  19*loop_expression0x8217fa0+10*loop_expression0x8217fa0^2
int foobar( int bound1, int bound2   )
  {
   for (int i = 0; i < bound1; i++)
     {
      array[i] = 0;
      for (int j = 0; j < bound1; j++)
        {
         x = 0;
        }
     }
   for (int i = 0; i < bound2; i++)
     {
      array[i] = array[i  -1] + array[i+1] ;
      for (int j = 0; j < bound2;    j++)
        {
         x = 0;
        }
     }
   return x;
  }
```




```
// Count -controlled loop complexity:
// 8+13*loop_expression0x8217  f78^3+4*loop_expression0x8217f78^4   +
int main()
  {
   for (int i = 0; i < bound; i++)
     {
       array[i] = 0;
      for (int j = 0; j < bound; j++)
        {
         x = foobar( bound,bound  );
        }
     }
   return 0;
  }
```



Unparsed Example

Preserves formatting, comments, and preprocessor control structure

Original Input C++ Source code

```
#include "A++.h"
#include "../include/ROSE_TRANSFORMATION_SOURCE.h"
#include <iostream.h>

int main() {

  int x = 4;

  //these comments are difficult
  for (int i = 0; i < 10; i++) {
    while (x) {
      x = x + 1;

      if (false) { x++; x = 7+x; }
      else {
        x = x - 1;
        x--;
      }

      // comments!
      x++;
      x += 1;
    }
  }
  return 0;
}
```

Unparsed Output C++ Source code

```
#include "A++.h"
#include "../include/ROSE_TRANSFORMATION_SOURCE.h"
#include <iostream.h>

int main(){

  int x=4;

  //these comments are difficult
  for (int i=0; i < 10; i++){
    while(x){
      x = x + 1;

      if (false){ x++; x = 7 + x; }
      else {
        x = x - 1;
        x--;
      }

        // comments!
      x++;
      x += 1;
    }
  }
  return 0;
}
```

Interactions with Others

- DOE Laboratories:
  - LLNL (A-Div (Kull), B-Div (IRS), Mark Graff, TSTT, Overture, Babel)
  - ANL (Paul Hovland)
  - ORNL
- DOE Research Programs:
  - PERC (SLAC, TSTT, C/C++ Optimization, UT, ANL, Dyninst Binary Rewriting)
- Collaborations:
  - IBM Haifa (Shmuel Ur)
  - Texas A&M (Lawrence Rauchwerger, Bjarne Stroustrup)
  - Rice University (Ken Kennedy, John Mellnor-Crummey)
  - Vienna University of Technology (Markus Schordan)
  - University of Tennessee (Jack Dongarra's group)
  - Cornell University (Sally McKee, Brian White)
  - Indiana University (Andrew Lumsdaine, Jeremiah Willcock)
  - University of California at Berkeley (UPC, Kathy Yelick)
  - University of Oslo (Hans, Andreas, Are)
  - University of Maryland (Jeff Hollingsworth, Chadd Williams)
  - Friedrich-Alexander-University Erlangen-Nuremberg (Markus Kowarschik, Nils Thurey)
  - University of Texas at Austin (Calvin Lin)
  - USCD (Scott Baden)
  - London Imperial College (Olav Beckman, Paul Kelly)
  - UC Davis (Su, Bishop)